

# AIアシスタント「たまちゃん」の進化： AIエージェント化への挑戦と、 システム開発用RAG「Omni-Q」への横展開

2025/11/11

デジタルプラットフォームサービスセンター

安井・濱野

※本資料に記載されているロゴ、システム名称、企業名称、製品名称は各社の登録商標または商標です。

EXcited About *The Future* .

デジタル革新で“ワクワク”する未来を共に創造します

1. これまでのたまちゃん開発の取り組み
2. AIエージェント版たまちゃんのご紹介
3. システム開発用のRAG「Omni-Q」のご紹介

## これまでのたまちゃん開発の取り組み

- ◆ 生成AIを活用し、エクサの社内情報を基に質問応答を行うSlackチャットボットです。
- ◆ 現在、7つの部署の情報に対応しています。(9/30に管理部の情報を追加しました)
  - 10月の利用実績では、人事部関連の質問が最も多く、次いで総務部、管理部が続く結果となりました
- ◆ 「@たまちゃん」とメンションをつけることで、生成AIとの会話が可能です。
- ◆ 全社公開の「#help-たまちゃん」での利用と**個人のプライベートチャンネルでの利用が可能です。**
- ◆ 検索拡張生成（RAG）の仕組みを採用しています。



安井 由香 たった今

@たまちゃん 名刺を緊急で明日までに用意したいです。どのように手続きをすればいいですか？

件の返信



たまちゃん アプリ たった今

お急ぎの場合は、通常名刺だと納期が最短10稼働日前後（＝明日では間に合わない）となるため、経費削減の一環で推奨している「簡易名刺」での依頼がおすすめです。簡易名刺は最短1時間程度で作成できるので、明日までに必要な場合はこちらをご利用ください。

依頼方法は以下のいずれかになります。

- ① お手持ちの名刺に直接訂正箇所を書き込み、Concierge Desk（アイマークプレイス2F）へ直接持参する
- ② 訂正内容を記入した名刺の現物をPDF化し、Concierge Desk宛にメール送信する
- ③ 必要事項（種類（簡易名刺－10枚単位）、部署名・氏名、肩書、もしあれば資格情報、住所、電話番号など）を本文に記載し、Concierge Desk宛にメール送信する

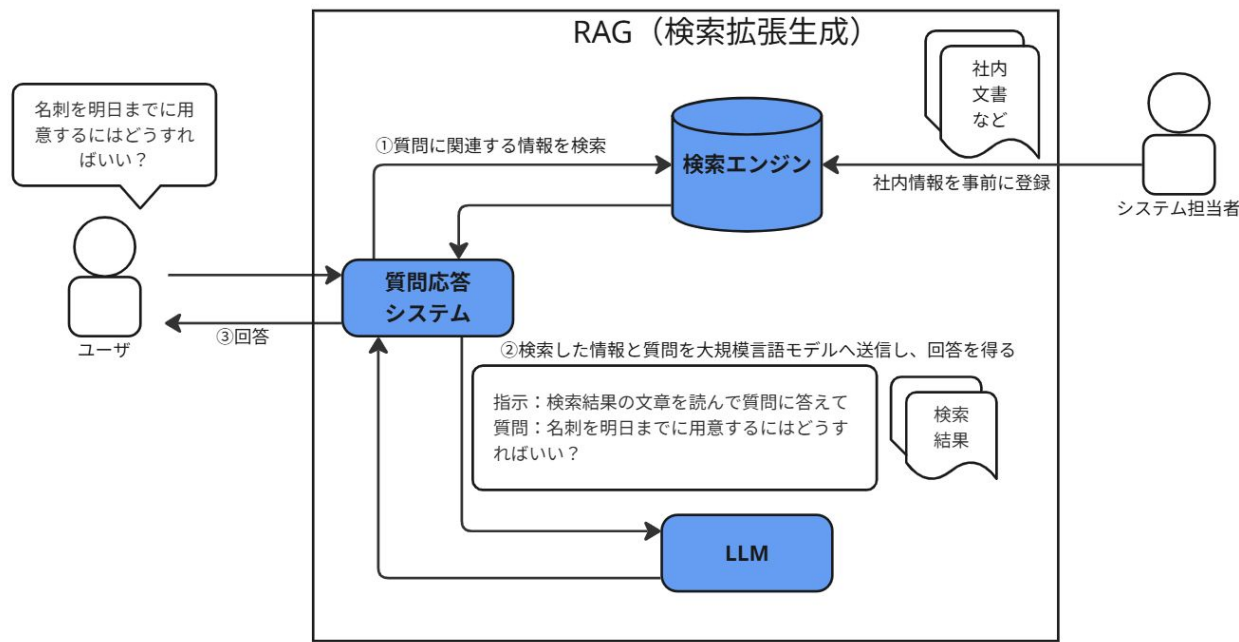
※依頼の際は、急ぎであること（明日までに必要）を必ず記載してください。なお、自宅への郵送も可能ですが、通常普通郵便の場合2～3日かかりますので、急ぎの場合は直接お取りに来る方法をおすすめします。

参考：<https://sites.google.com>

参考情報のURL

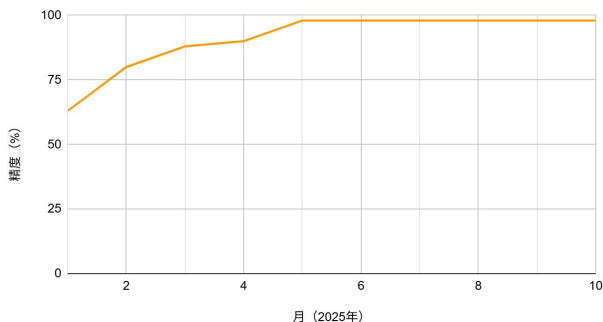
# 検索拡張生成（Retrieval Augmented Generation・RAG）とは *exa* EXA CORPORATION

- ◆ 検索拡張生成（RAG）は、ユーザの質問に関連する文章を検索し、その情報を大規模言語モデル（LLM）への入力（プロンプト）に加えて回答を生成する技術です
- ◆ RAGにより、LLMの学習データにはない社内情報などを参照し、より正確な回答が可能となります

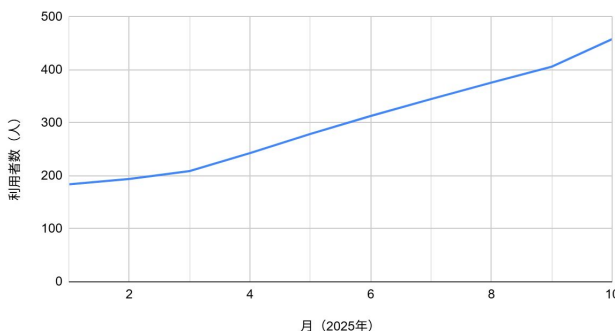


- ◆ たまちゃんはリリース直後、精度（回答ができない頻度）に課題があり、利用者増加に伸び悩んでいたため、精度やUI/UXの改善活動を2025/2～7に実施しました
- ◆ 精度改善施策を11個、UI/UX改善施策を3つ実施しました（詳細はKnowledgeやエクサHPのコラムをご参照）
  - 2025/2～4
    - 施策を適用し始めてすぐに精度が劇的に向上
    - 利用者数と利用回数が急激に増加
  - 2025/2～10
    - 利用者数：約2.4倍
    - 利用回数：約7.7倍
  - 2025/9～10
    - 新たに2部署の情報を追加
    - 利用者数と利用回数が増加

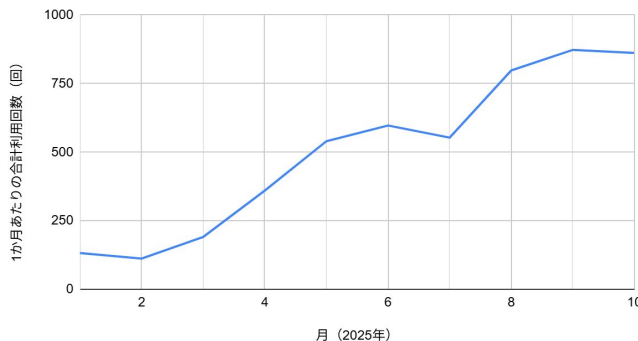
精度（回答できた件数の割合）の推移 2025/01～2025/10



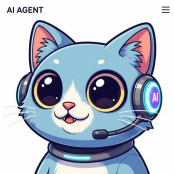
利用者数の推移 2025/01～2025/10



1か月あたりの合計利用回数の推移 2025/01～2025/10



これまでのたまちゃんをさらに発展させるため、2つの取り組みを行っています



## AIエージェントへの進化

より高度な指示に対応できるよう、  
自律的にアクションを実行する



## システム開発の現場へ

RAG技術を応用し、システム開発で  
利用できる「Omni-Q」を開発

次ページからこれらの取り組みについて詳しくご紹介します

# AIエージェント版たまちゃんのご紹介



従来の「たまちゃん」は限定的なアクション（決められた手順で検索を実行して回答）しか行えない制約があったが、「AIエージェント版たまちゃん」では、AIエージェントの技術を活用することで、ユーザの高度な指示に応じられるか検証

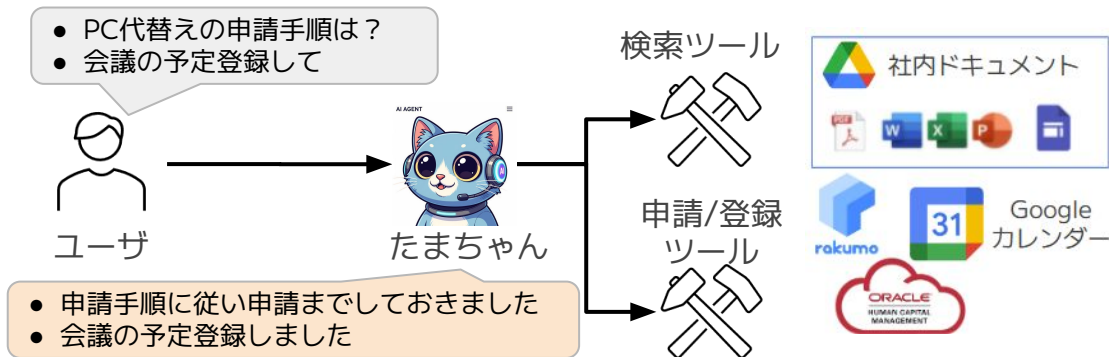
## 従来のたまちゃん

決められた手順で、  
社内ドキュメントを検索して回答



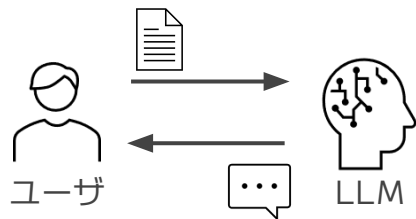
## AIエージェント版のたまちゃん

- 自律的に必要な検索ツールを実行して得られた総合的な情報を元に回答（≒Agentic RAG）
- 自律的に手続きの申請や登録用のツールを実行することで、手続きをサポート



高度な推論能力と自然言語理解能力を持ったLLMをAIエージェントの頭脳として機能させることで、AIエージェントは要求に対して、自律的に計画を立てて、アクションを実行できる

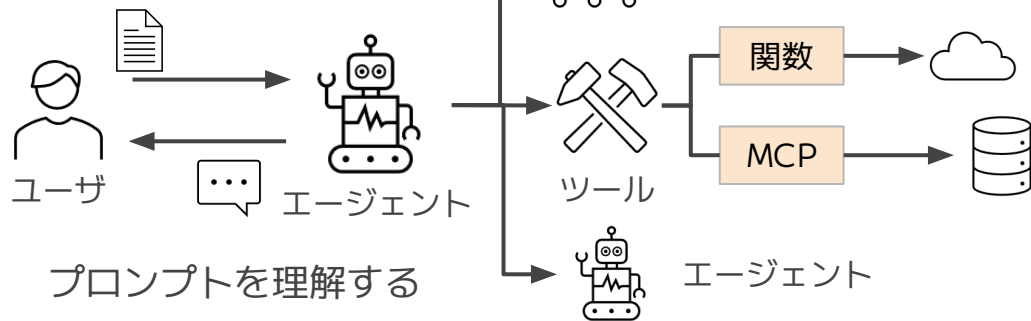
## 従来のLLMの使い方



1. プロンプトを理解する
2. 回答する

プロンプトエンジニアリングが重要

## AIエージェント



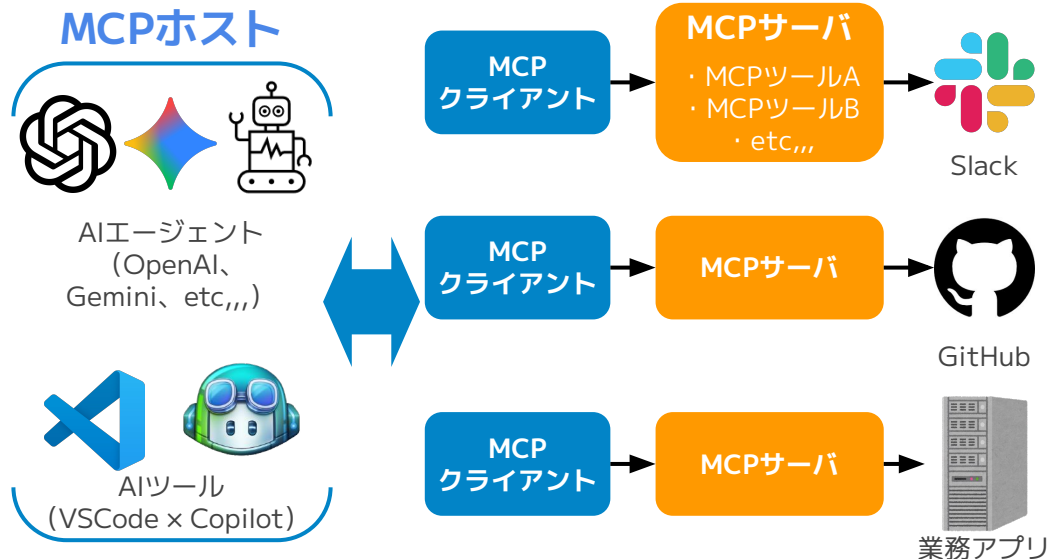
1. プロンプトを理解する
2. 計画を立てる
3. アクション（ツールの実行や、他のエージェントと協調）を実行
4. 回答する

コンテキストエンジニアリング（プロンプト、ツール、メモリ等）が重要

MCP (Model Context Protocol) とは、Anthropic社が定めたAIエージェントとツールの接続方法に関する標準規格のことで、AIエージェントとツールの連携が容易に

## MCPの規格で接続する流れ

様々なMCPホストとMCPサーバが柔軟に切り替え可能

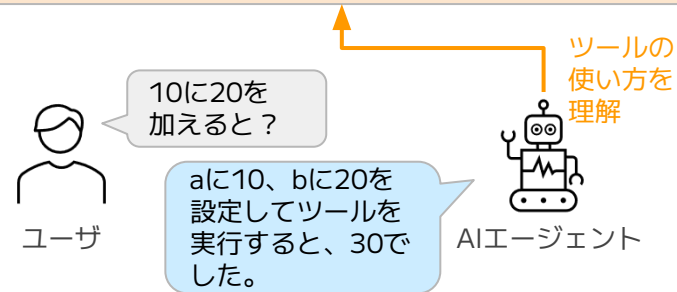


## MCPツールの実装例

### AddNumbersTool (int a, int b)

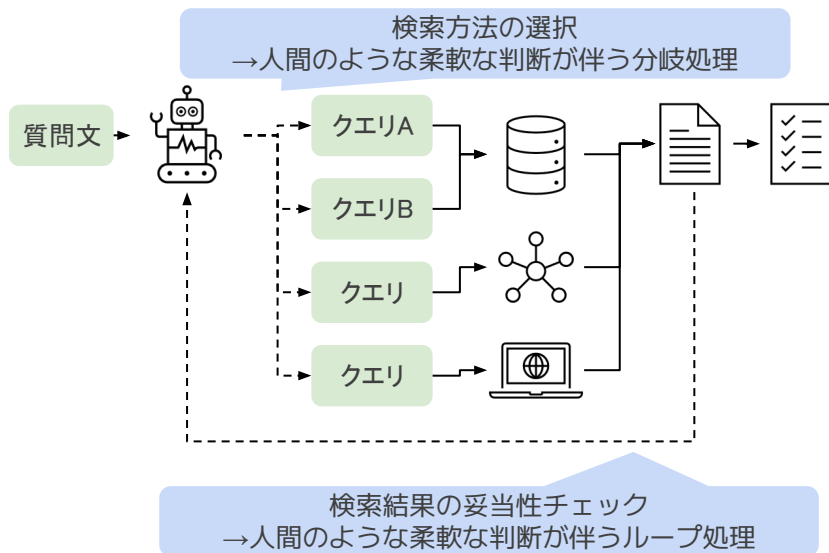
以下のアノテーション (説明文) をあらかじめ付与

- ツールの説明文
  - 「このツールは2つの数を計算します」
- 引数の説明文
  - 「a: int型で基準になる数を設定してください。」
  - 「b: int型で加えたい数を設定してください」

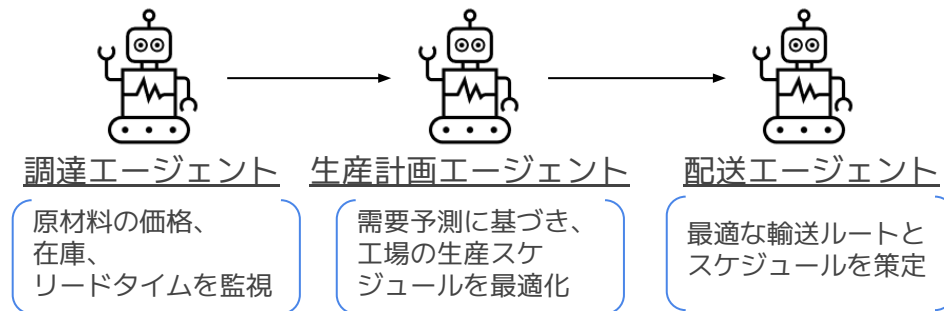


従来ではシステム化が難しかった自由度の高いフローや、役割のある人物同士が協力するフローの実装に役立つ

## 自由度の高いフロー（例:Agentic RAG）



## 役割のある人物同士が協力するフロー （例：サプライチェーン最適化）



2つの方向性でAIエージェントを技術検証中

	「検索」を実行するエージェント	「登録」を実行するエージェント
特徴	<ul style="list-style-type: none"><li>読み取り専用（Read-Only）の操作を実行</li><li>実装のハードルが比較的低い</li></ul>	<ul style="list-style-type: none"><li>書き込み操作（Write）を実行</li><li>実装のハードルが比較的高い</li></ul>
考慮点	<ul style="list-style-type: none"><li>検索コンテンツの拡充</li><li>検索の精度</li></ul>	<ul style="list-style-type: none"><li>データの整合性</li><li>エラーハンドリング</li><li>認証/認可</li></ul>
取り組み事例	社内コンテンツ検索エージェント	<ul style="list-style-type: none"><li>（断念※）<del>RakumoWF申請エージェント</del></li><li>会議調整エージェント</li></ul>

※RakumoWF申請を行うエージェントも検証しようとしたが、管理者用のAPIキーでは代理申請（下書きが代理申請者に保存されてしまう。申請後の内容編集ができない。）となる制約があったため断念

社内コンテンツ検索エージェントでは、社内ドキュメント検索が可能なツールに加え、RakumoWFの申請書の情報を検索するツールを設定することにより、RakumoWF関連の質問に対して高度な回答ができるか検証

## 従来のたまちゃん

決まった手順で社内ドキュメントを検索して回答

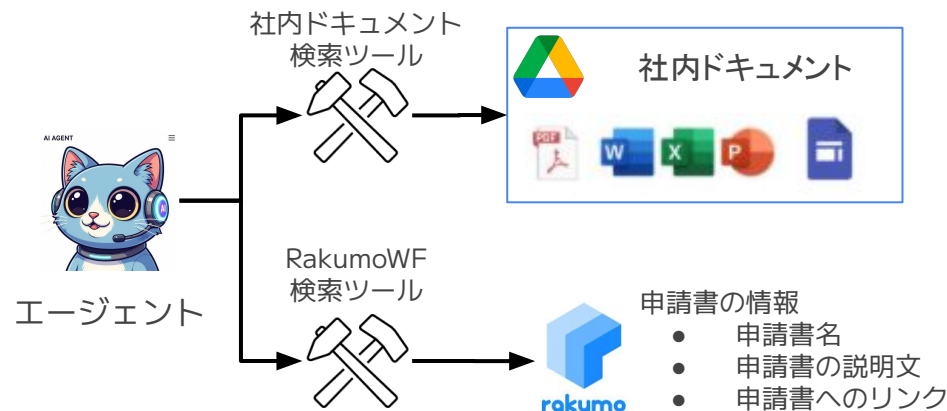


たまちゃん



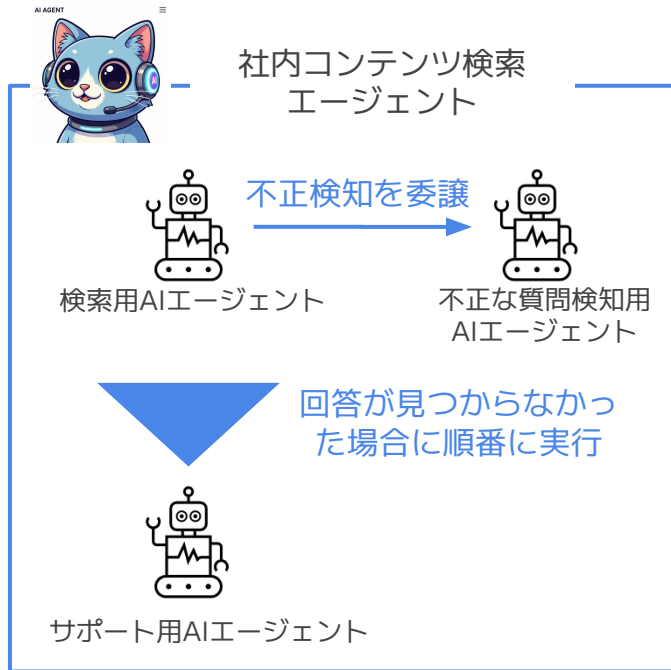
## 社内コンテンツ検索エージェント

エージェントが臨機応変にツールを実行して回答



社内コンテンツ検索エージェントは、3つのエージェントで構成されており、検索用AIエージェントには、なるべく多くの検索ツールを実行するよう指示

## エージェントの内部



## 検索用AIエージェントのプロンプト (一部のみ抜粋)

あなたはIT会社の社内ヘルプデスクAI「たまちゃん」です。  
ユーザーからの質問に対して、#回答作成のルール に従い、回答してください。  
回答が見つからなかった場合は無理に回答を作らず、#回答できなかった場合のルール に従ってください。

# 回答作成のルール

1. 利用可能な検索ツールを活用し、情報を調査してください。

- 検索可能なツールは積極的に利用し、1つのツールに偏らず、複数のツールから得た情報を総合して回答を作成してください。

- ある検索ツールの実行結果や取得した情報をもとに、追加で他のツールによる調査や情報取得が有効・必要だと判断した場合は、自律的に必要なツールを続けて実行し、その結果を回答に反映してください。

(例：ドキュメント検索の結果、さらに詳細な情報や関連する申請書の情報が必要と考えられる場合など)

# 回答が見つからなかった場合のルール

1. 質問に関連する情報を見つけられなかった場合は、無理に推測せず、下記の内容を含めて回答してください。

- 「{質問}」に関する情報が見つからなかったこと
- 「すみません」「ごめんなさい」など、謝罪の言葉を必ず含めること

なるべく多くの検索ツールを実行して総合的な回答をするよう指示

社内コンテンツ検索エージェントには、2つの検索ツール（社内ドキュメント検索ツール、RakumoWF検索ツール）をMCPツールとして設定



## 社内ドキュメント 検索ツール

- ツールの説明文
  - 質問文を元に社内ドキュメントを検索する
- 引数の説明文
  - question：ユーザからの質問文
  - messages：ユーザとAIの過去の発話履歴
  - top：検索結果の上位何件までを取得するか



## RakumoWF 検索ツール

- ツールの説明文
  - 検索クエリに該当する申請書を検索する。該当する申請書が見つからなかった場合は、空のリストを返す。
- 引数の説明文
  - query：ワークフロー検索用の検索クエリ

これらの説明文を元に、エージェントは検索ツールの実行タイミングや実行方法を理解する

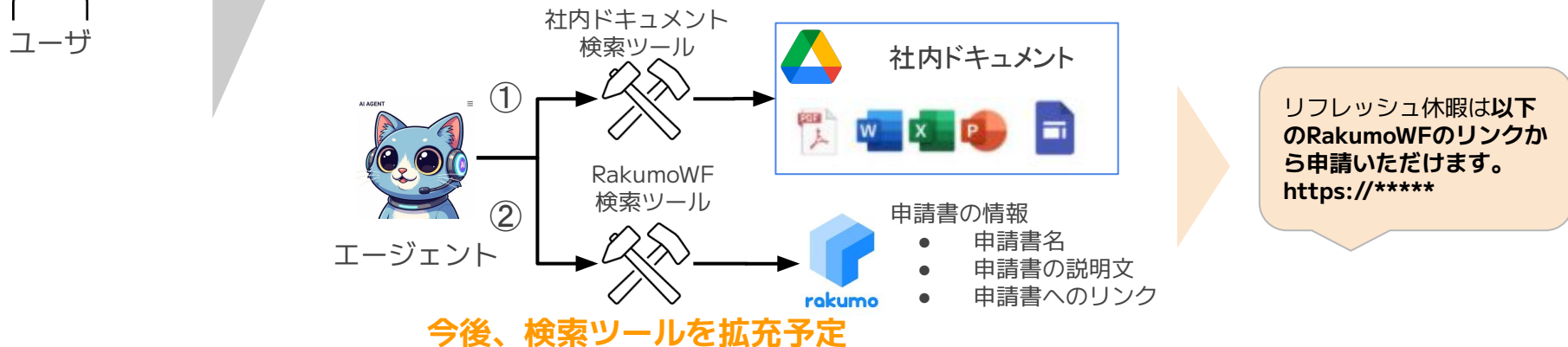


社内コンテンツ検索エージェントは、従来のたまちゃんに比べ高度な回答（WFの補足情報が付与された回答）ができることが分かった

## 従来のたまちゃん



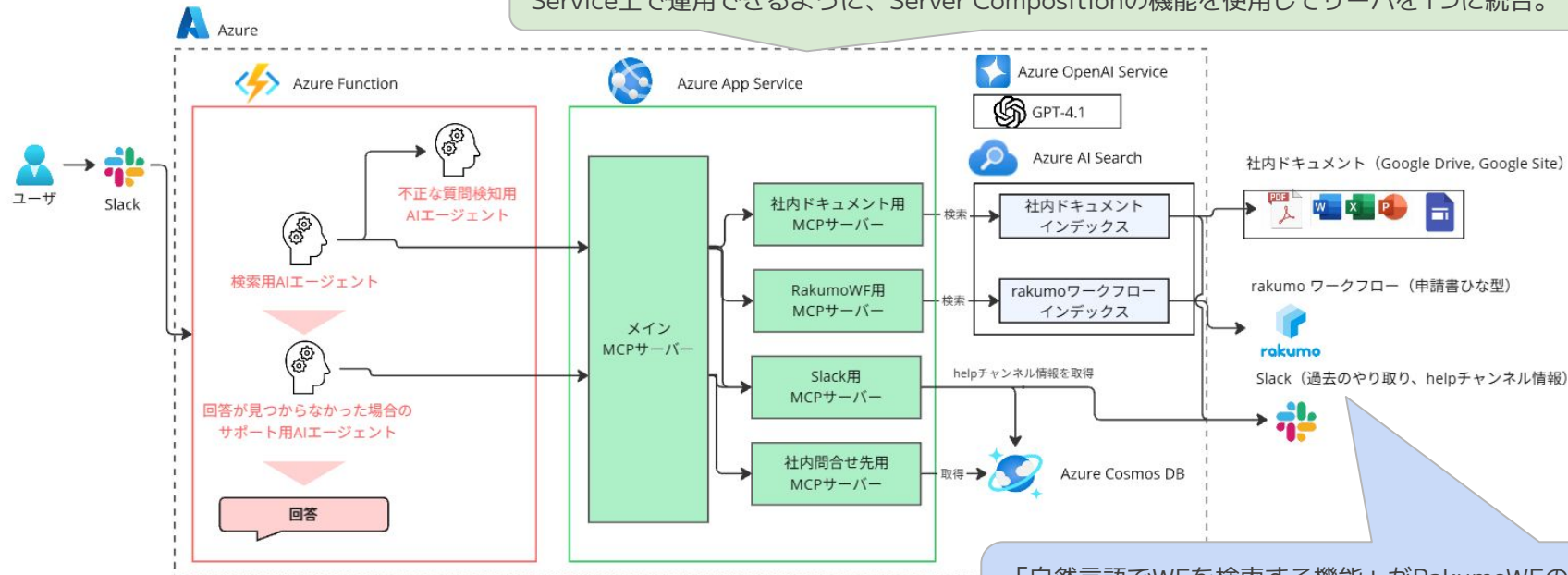
## 社内コンテンツ検索エージェント



今後、検索ツールを拡充予定

## 社内コンテンツ検索エージェントは、Azure上に以下のアーキテクチャで実装

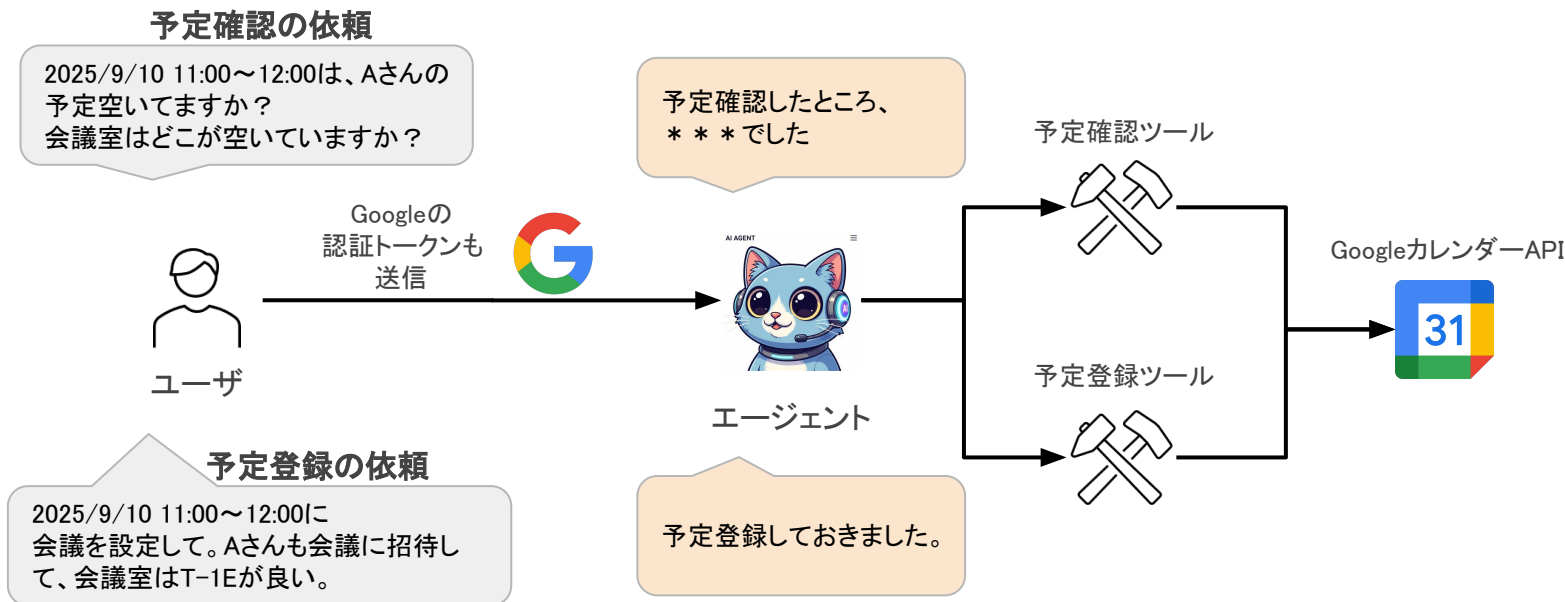
MCPサーバはFastMCPを使用して実装。MCPサーバの管理を楽にするため、1つのAzure App Service上で運用できるように、Server Compositionの機能を使用してサーバを1つに統合。



AIエージェントは、OpenAI Agents SDKを使用して実装

「自然言語でWFを検索する機能」がRakumoWFの標準のAPIには存在しなかったため、WFの情報を全文検索エンジンに登録し、MCPツールに検索機能を実装

会議調整エージェントでは、GoogleカレンダーのAPIを実行して予定確認や予定登録を行うエージェントを検証中



## 会議調整エージェントでは、予定が空いているか確認するケースと予定を登録するケースで、行ってほしい手順を指示

### プロンプト

あなたは、Google Calendarと連携して、会議の予定確認や予定登録をサポートするAIエージェントです。  
ユーザーの要望に応じて、提供されたツールを適切に利用し、自然な対話を通じてタスクを遂行してください。  
ユーザーから予定確認を依頼された場合は、「予定が空いているか確認する場合」のルールに従ってください。  
ユーザーから予定登録を依頼された場合は、「予定を登録する場合」のルールに従ってください。

#### # 予定が空いているか確認する場合

予定が空いているか確認するツールがあるかと思いますが、必要なパラメータがユーザの発話に含まれていなかった場合、対話形式でユーザーから聞き出してから、実行してください。

日時を聞き出す際は、今日、明日ではなく、具体的な日付（例：2024年1月1日）で聞き出してください。

全員の予定が空いていた場合のみ、利用可能な会議室があれば、会議室もユーザーに提示してください。

#### # 予定を登録する場合

予定を登録するツールがあるかと思いますが、必要なパラメータがユーザの発話に含まれていなかった場合、対話形式でユーザーから聞き出してから、実行してください。

日時を聞き出す際は、今日、明日ではなく、具体的な日付（例：2024年1月1日）で聞き出してください。

予定登録のツールを実行する直前では、ユーザーに登録内容の確認を行い、ユーザーからの承認を得てから実行してください。

予定を登録するとツールから、Google Calendarの予定詳細画面へのリンクが得られると思いますが、そのリンクをユーザーに提供してください。

エージェントがユーザと対話しながら正しい順番で処理を進めていけるように  
処理の手順や処理の内容を明確に指示

会議調整エージェントには、2つのツール（予定確認ツール、予定登録ツール）をMCPツールとして設定



## 予定確認ツール

- ツールの説明文
  - 指定された時刻の範囲内において、Googleカレンダーに予定が登録されていないかチェックする。  
一緒に予定を確認したいユーザのメールアドレスが指定されていた場合は、そのユーザの予定も確認する。また、利用可能な会議室が存在する場合は、その会議室の詳細情報を返す。
- 引数の説明文
  - start\_time、end\_time：予定の検索範囲の開始/終了時刻（ISO 8601形式。タイムゾーンはAsia/Tokyo。例: 2025-08-25T09:00:00+09:00）
  - another\_user\_emails：一緒に予定を確認したいユーザのメールアドレスのリスト



## 予定登録ツール

- ツールの説明文
  - Googleカレンダーに予定を登録する。予定に加えたユーザのメールアドレスが指定されていた場合は、そのユーザを予定の参加者として登録する。会議室のresourceEmailが指定されていた場合は、その会議室を使用する設備として登録する
- 引数の説明文
  - start\_time、end\_time：予定の検索範囲の開始/終了時刻（ISO 8601形式。タイムゾーンはAsia/Tokyo。例: 2025-08-25T09:00:00+09:00）
  - title：予定のタイトル
  - description：予定の説明
  - attendees：予定に招待したい人のメールアドレスのリスト
  - conference\_room\_resource\_email：会議室のresourceEmail

これらの説明文を元に、エージェントはツールの実行タイミングや実行方法を理解する

## 会議調整エージェントに予定確認を依頼した時の実行例は以下の通り



ユーザ

予定を確認したい

ご予定を確認するために、具体的な日付と時間帯を教えてくださいませんか？

2025年9月3日 10:00～11:00に自分の予定が空いているか確認したい

2025年9月3日 10:00～11:00のご予定は空いています。また、利用可能な会議室として「T-3E」（座席数4）もご利用いただけます。会議室の予約や他の方の予定確認も必要な場合はお知らせください。

{他の人のメールアドレス}の予定も空いているか確認したい

{他の人のメールアドレス}さんも予定空いています。会議室の予約や他の方の予定確認もご希望でしたらお知らせください。

2025年9月3日 13:00～14:00はどうですか？

2025年9月3日 13:00～14:00の時間帯は、あなたと{他の人のメールアドレス}さんの両方に予定が入っており、空いていません。

←ツール実行に必要な  
パラメータの値をヒアリング

AI AGENT



エージェント

←過去の文脈も理解しつつ、  
他のユーザの予定が  
空いているかも確認して回答

## 会議調整エージェントに予定登録を依頼した時の実行例は以下の通り



ユーザ

予定を登録したい

予定の登録ですね。いくつか確認させてください。  
1. 予定のタイトル（例：営業会議など）  
2. 日付と開始時刻・終了時刻（例：2024年7月10日  
14:00～15:00）  
3. 参加者のメールアドレス（招待したい方がいれば）  
4. 会議室の利用希望があれば、会議室名  
5. 予定の説明（任意）  
上記について教えていただけますか？

タイトルは「動作確認用の予定登録」、予定の日時は  
「2025/9/4 9:00～10:00」、招待したい人は{他の人のメール  
アドレス}、会議室はT-1E

ありがとうございます。内容を確認させてください。

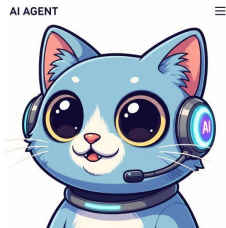
- ・タイトル：動作確認用の予定登録
- ・日時：2025/9/4 9:00～10:00
- ・参加者：{他の人のメールアドレス}
- ・会議室：T-1E

予定の説明を追加したい場合はご記入ください。特に無ければこの  
まま登録してもよろしいでしょうか？

登録してください

予定を登録しました。  
Googleカレンダーで詳細を確認したい場合は、下記リンクからご覧  
いただけます。  
[https://\\*\\*\\*](https://***)

←ツール実行に必要な  
パラメータの値をヒアリング

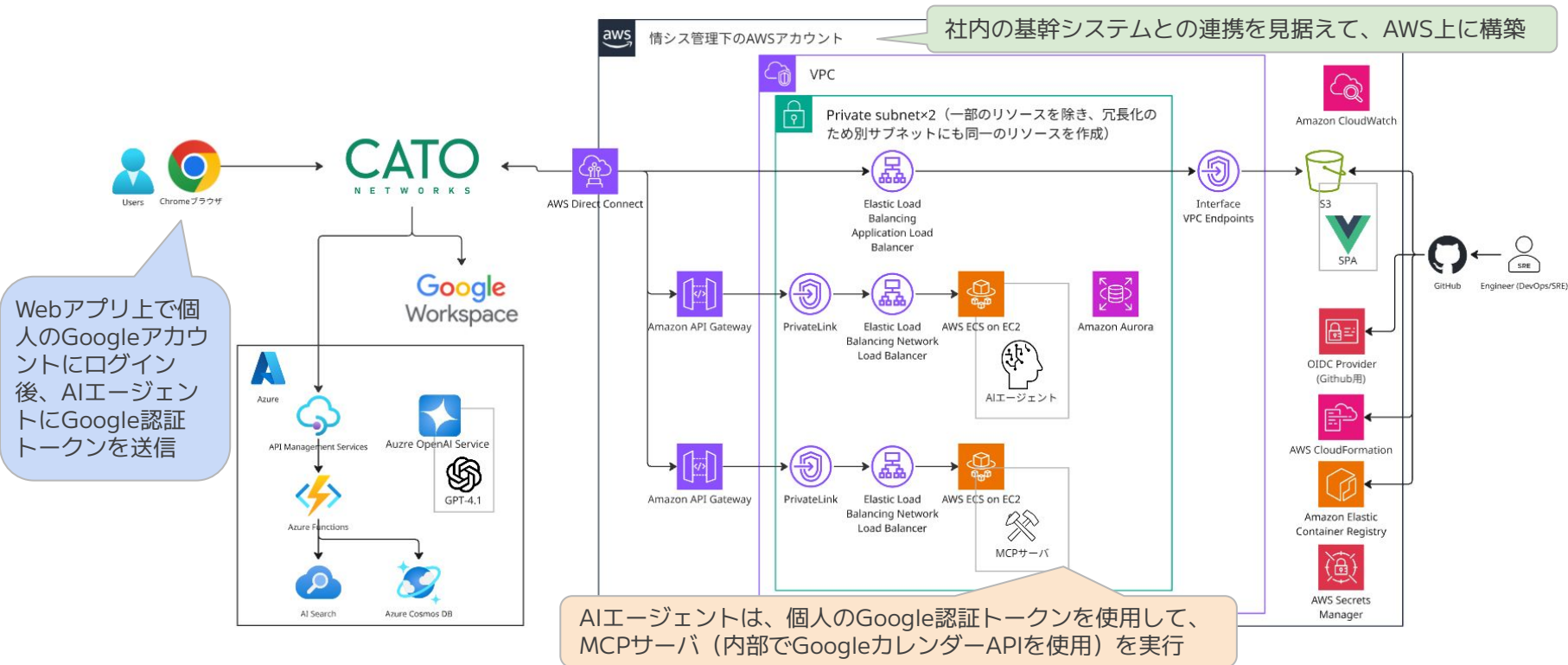


エージェント

←登録内容を復唱して  
ユーザに確認

←登録したカレンダーの  
リンクを提示

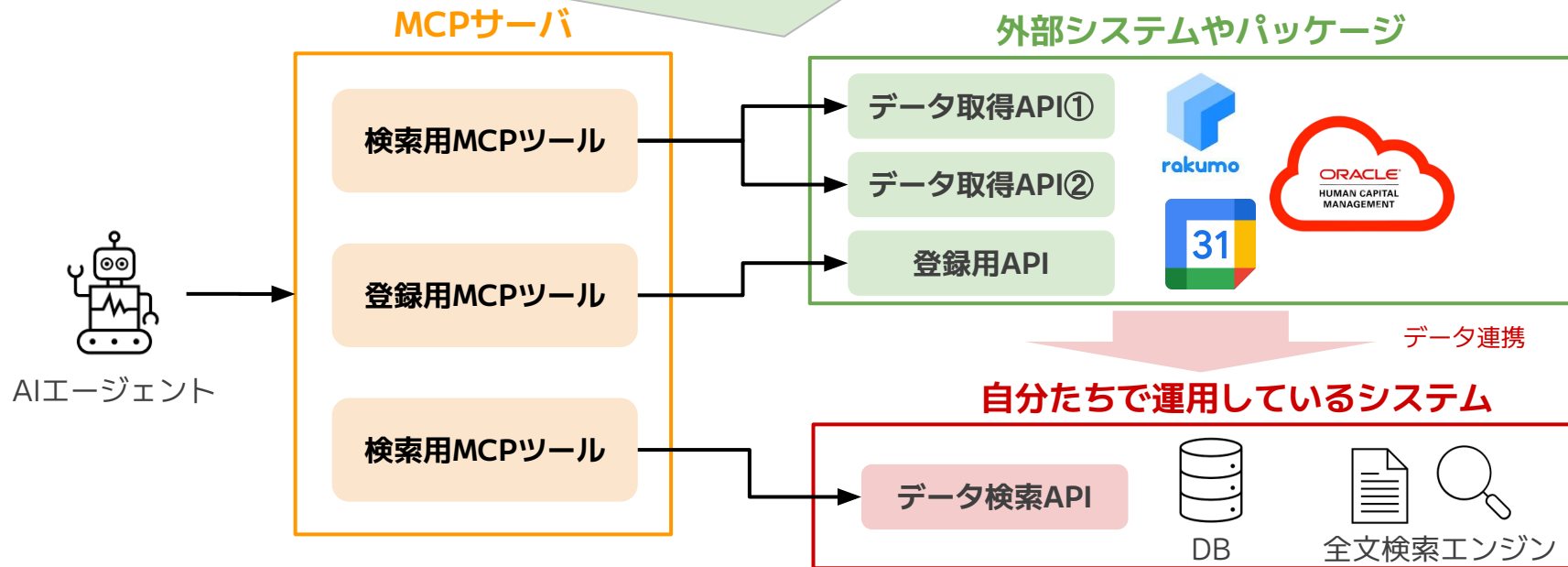
## 会議調整エージェントは、GoogleカレンダーAPI実行の都合上、Googleアカウントにログイン可能なWebアプリケーションの形式で、AWS上に構築





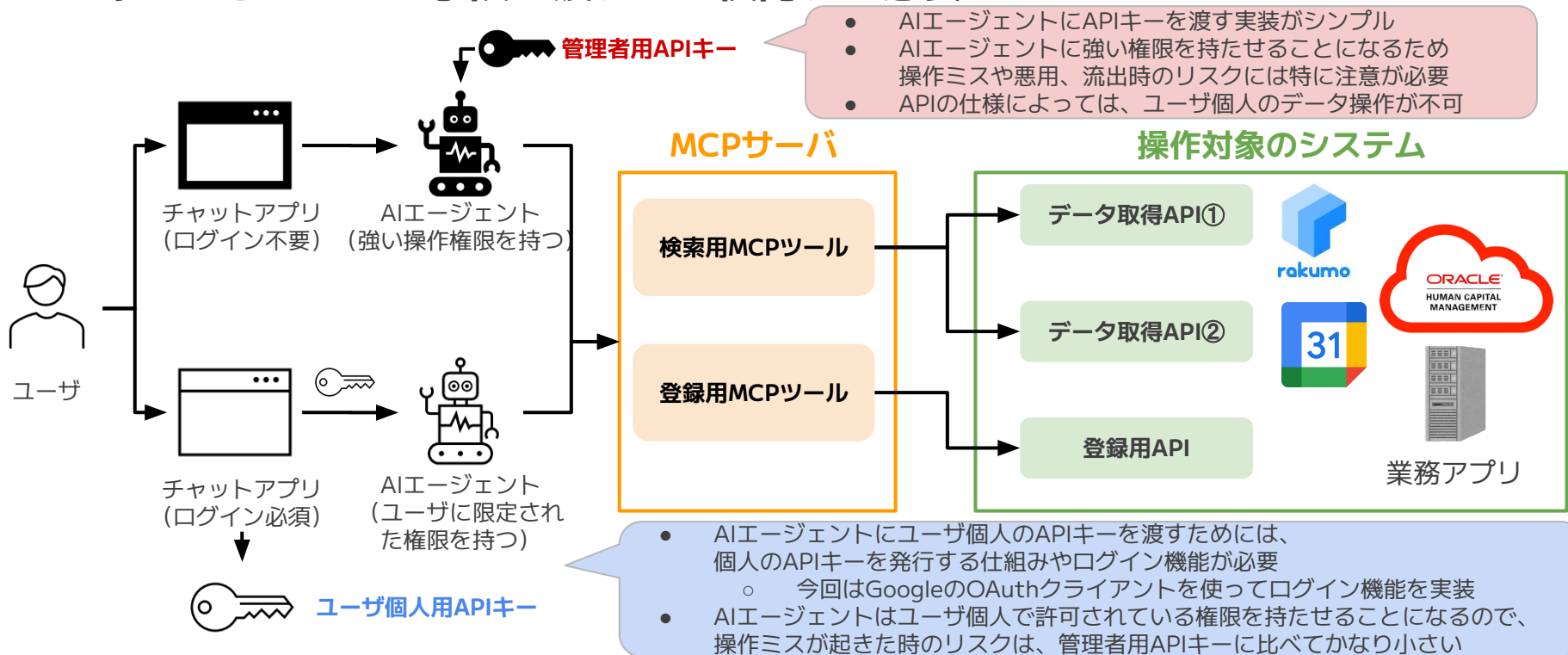
## 外部システム（RakumoWF、Googleカレンダー、Oracle HCMなど）をAIエージェントと連携させる場合、提供されているAPIをラップしたMCPサーバを構築するのがおすすめ

外部システムの公式MCPサーバが提供されていない場合、外部システムのAPIをラップしたユースケース特有のMCPサーバ構築がおすすめ。MCPツールへのアノテーションの工夫も可能

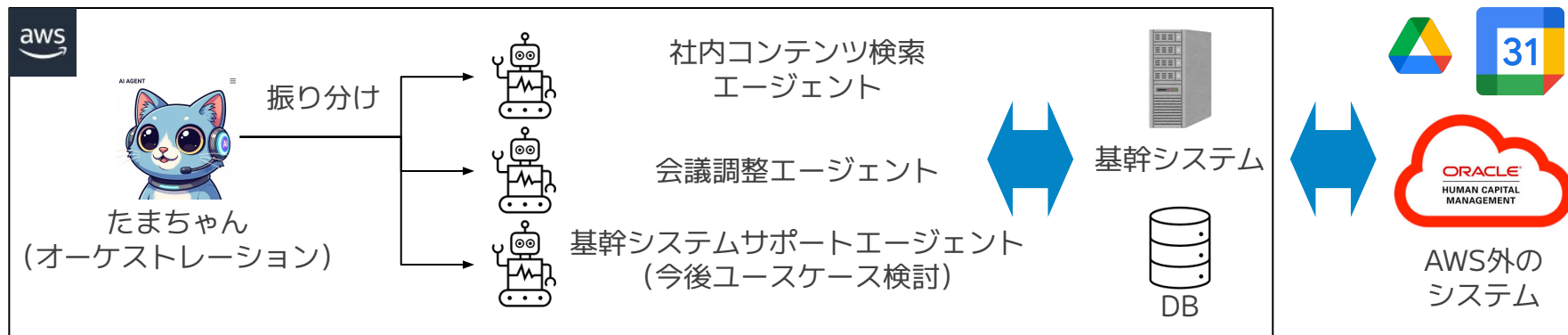


外部システムのAPI仕様がユースケースに適合しづらい場合（今回ではWF検索のケース）は、自分たちで運用しているシステムにデータ連携を行うことで解消出来ないか検討

## AIエージェントにシステムのAPIを実行させる場合、AIエージェントにどの種類のAPIキーをどのような手段で渡すかを検討する必要がある



- ◆ 社内コンテンツ検索エージェントと会議調整エージェントを1つのたまちゃんとして統合
  - 問い合わせ内容に応じて、検索または会議調整を臨機応変に実行
- ◆ 社内の基幹システムへの検索や登録作業をサポートするエージェントやMCPサーバ構築
  - 会議調整エージェントを構築したAWS環境内で稼働している様々な基幹システムと連携



## システム開発用のRAG「Omni-Q」のご紹介

# たまちゃん発展の取り組み② プロジェクト用RAGとしての横展開 EXA CORPORATION

- ◆ エクサではGeminiやNotebookLM、GitHub Copilotなどの生成AIツールが持ち帰り開発で利用可能になった
- ◆ 特にNotebookLM (RAG) は、若手SEの立ち上げに有効との報告が出てきているが、ファイル数上限などの制約から大規模なシステムへの適用は難しい

社内RAG「たまちゃん」で培ったRAGのノウハウをSEの業務支援へと展開

こんなエラーがログに出てたんだけど、過去に同様の事例あった？

プロジェクト  
メンバー

過去の記録によると、このモジュールの設定を変更して解決したようです。参考：[リンク](#)

Omni-Q



- ソースコード
- 設計書
- 議事録
- 障害履歴 など

ビジネス  
オーナー  
(業務知識)

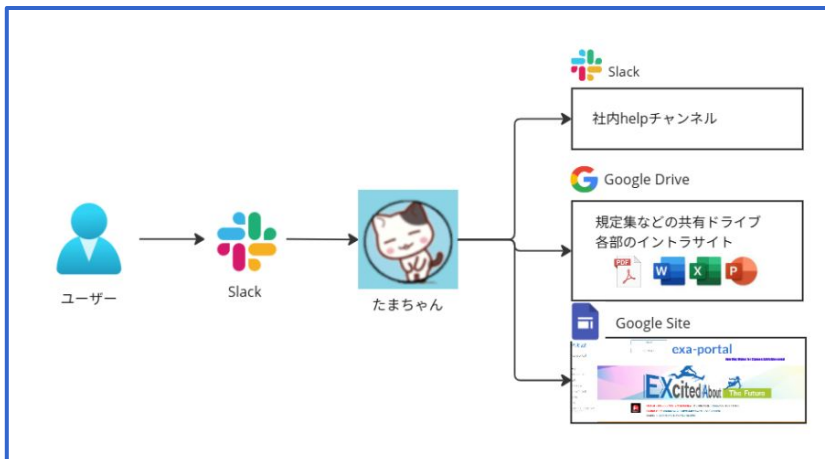
ベテランSE  
(開発・運用ナレッジ)

## Omni + Q

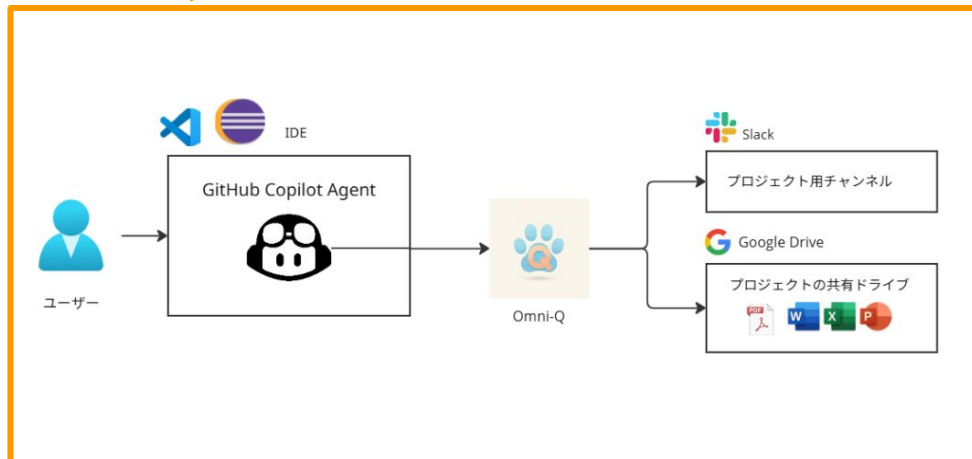
"Omni-" (あらゆる、すべての) と "Q" (Question/Query) の組み合わせ。プロジェクトのあらゆる情報源を基に質問に回答する

- ◆ SEの業務支援に特化した質問応答エージェント（RAGを活用）
- ◆ 情報源は現在GoogleDriveとSlackに対応（たまちゃんの仕組みを流用）
- ◆ たまちゃんと同じ管理Webからプロジェクト代表者が情報を登録する
- ◆ 利用者は開発用のIDE(Visual Studio CodeもしくはEclipse) のGitHub Copilotを通じて利用する
  - 開発者が使い慣れたIDEのUI/UXをそのまま活用できる点、コーディングのコンテキスト（開いているファイルやエラーログ）をシームレスに連携できる点でGitHub Copilotとの組み合わせを採用
  - 現在IDE以外からの提供も検討中（後述）

## 従来のたまちゃん



## Omni-Q



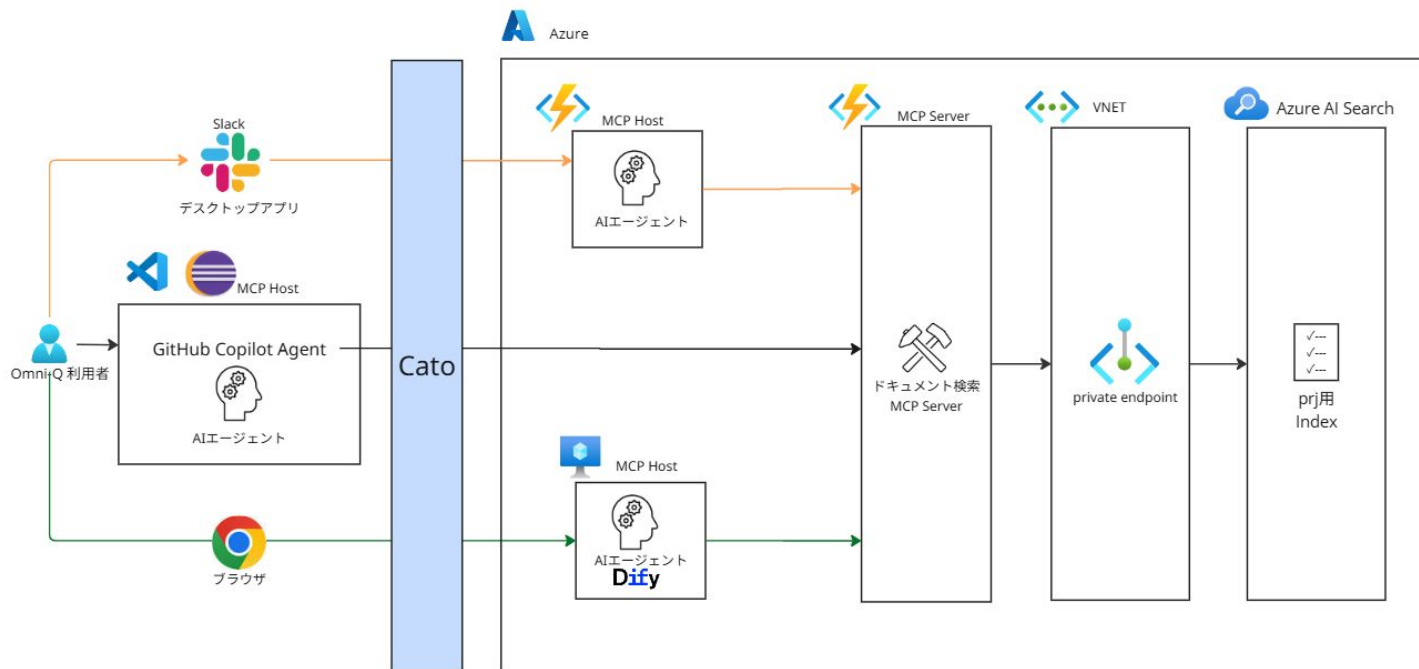
# Notebook LMとOmni-Qの比較

Notebook LMは小規模なプロジェクトで手軽に利用可能。ファイル数が多い場合やMS Office系のドキュメントがメインである場合はOmni-Qの利用を検討いただく

	NotebookLM Pro	Omni-Q
情報源	Google Drive 、 <b>Web URL</b> 、 アップロードしたファイル	Google Drive (共有ドライブ), <b>Slack (チャンネル)</b>
対応ファイル	PDF, Google系ドキュメント (スライド、ドキュメント、スプレッドシート) , テキスト、マークダウン	<b>Word, Excel, PPT</b> , PDF, Google系ドキュメント (スライド、ドキュメント、スプレッドシート) テキスト、マークダウン
ファイル数上限	1つのNotebookにつき <b>300ファイル</b> まで	Azure AI Searchのindexサイズ上限が15GB (*ファイルから抽出されたテキストデータや検索用データが格納される領域のサイズ)
登録データの更新	手動	<b>自動</b>
データの扱い	モデルの学習に利用しない	モデルの学習に利用しない
ナレッジ管理の方法	【個人 / チーム】 ユーザ個人がノートブックにソース ( <b>ファイル</b> ) 単位で手動追加。	【プロジェクト全体】 代表者が管理画面で <b>フォルダ単位</b> やチャンネル単位で一括登録。
利用インターフェース	Webブラウザ (専用UI)	<ul style="list-style-type: none"><li>・ IDE + GitHub Copilot</li><li>・ Webブラウザ (Dify)</li><li>・ Slack</li></ul>

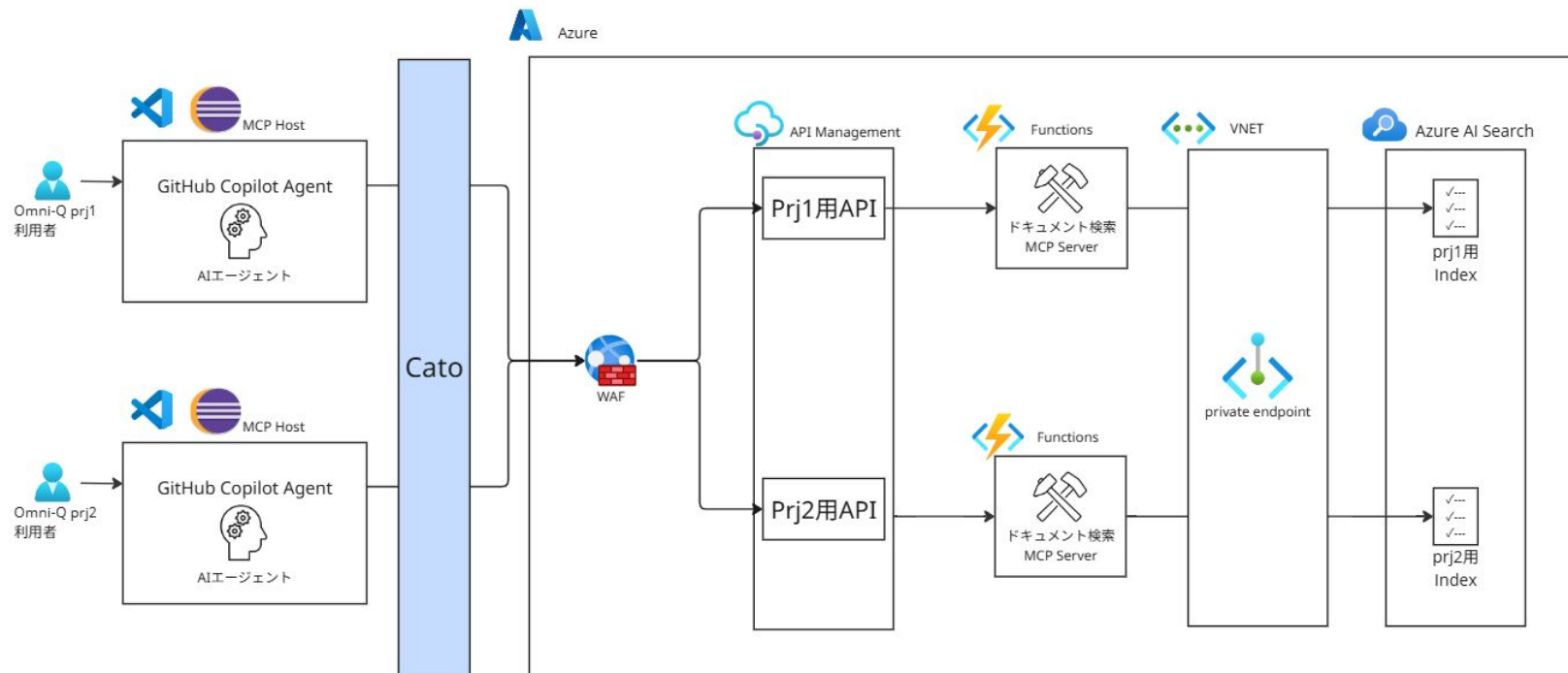
## ◆ IDE(Visual Studio CodeもしくはEclipse) のGitHub Copilot, Dify, Slackの3タイプの 利用方法を提供予定 (DifyとSlackは現在準備中)

- DifyはオープンソースのLLMアプリ開発プラットフォーム。ノーコードでLLMアプリを作成できる
- 詳細は昨年のEVF「[誰でもAIエージェントが作れる開発環境Dify/Llama3.1の紹介](#)」参照





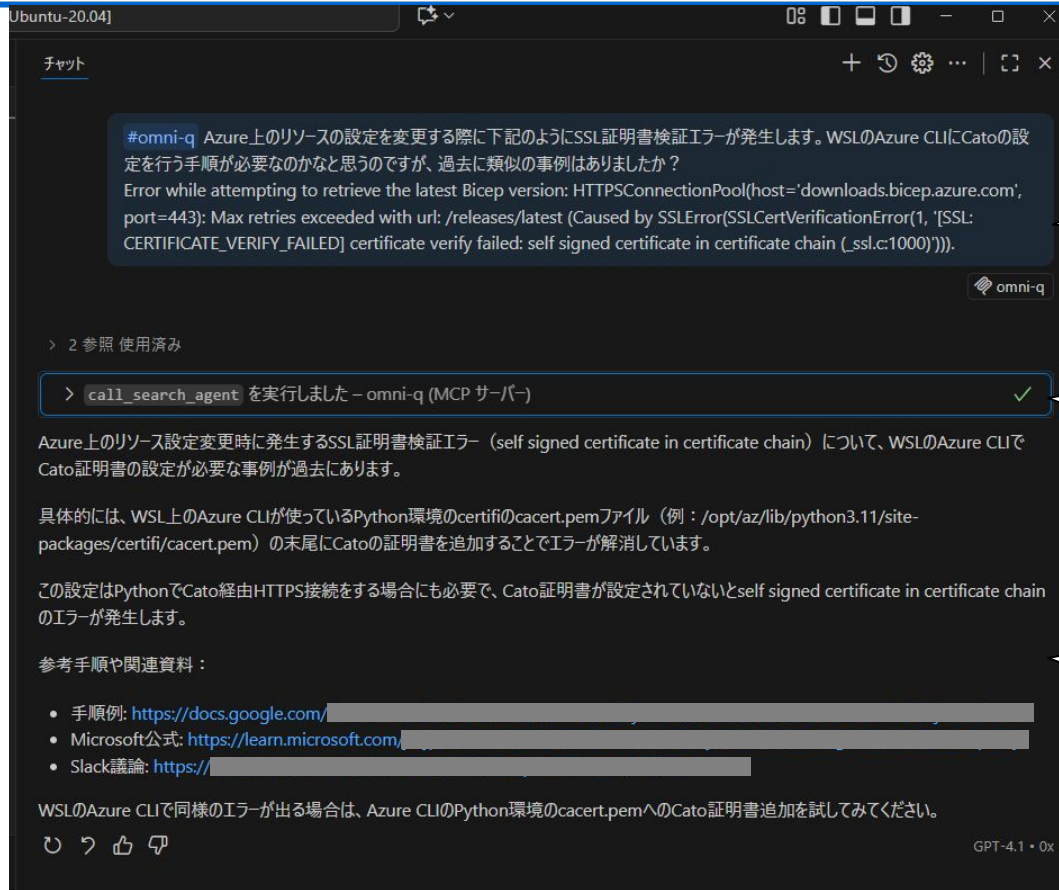
- ◆ AI Searchは全プロジェクトで共有し、indexをプロジェクト専用を用意
- ◆ MCP Serverはプロジェクト専用としてAzure Functionsにて作成



The screenshot displays the Visual Studio Code interface with the `azure_blob_log_uploader.py` file open in the editor. The file contains Python code for connecting to Azure Blob Storage. A white box labeled "コーディング中のエディタの画面" (Code editor screen) is overlaid on the code. On the right, the Chat window is open, showing a message from the `omni-q` agent regarding an SSL certificate error. A white box labeled "GitHub Copilotとのチャットを行う画面" (Screen for chatting with GitHub Copilot) is overlaid on the chat window. The chat window also shows a successful command execution: `call_search_agent` was executed - omni-q (MCP サーバー).

コーディング中のエディタの画面

GitHub Copilotとのチャットを行う画面



エラーログと共に質問内容を送信

GitHub CopilotがOmni-Qへ  
問い合わせ

Omni-Qからの回答を提示  
たまちゃんと同様に参考リンクから  
ドキュメントなどを確認できる

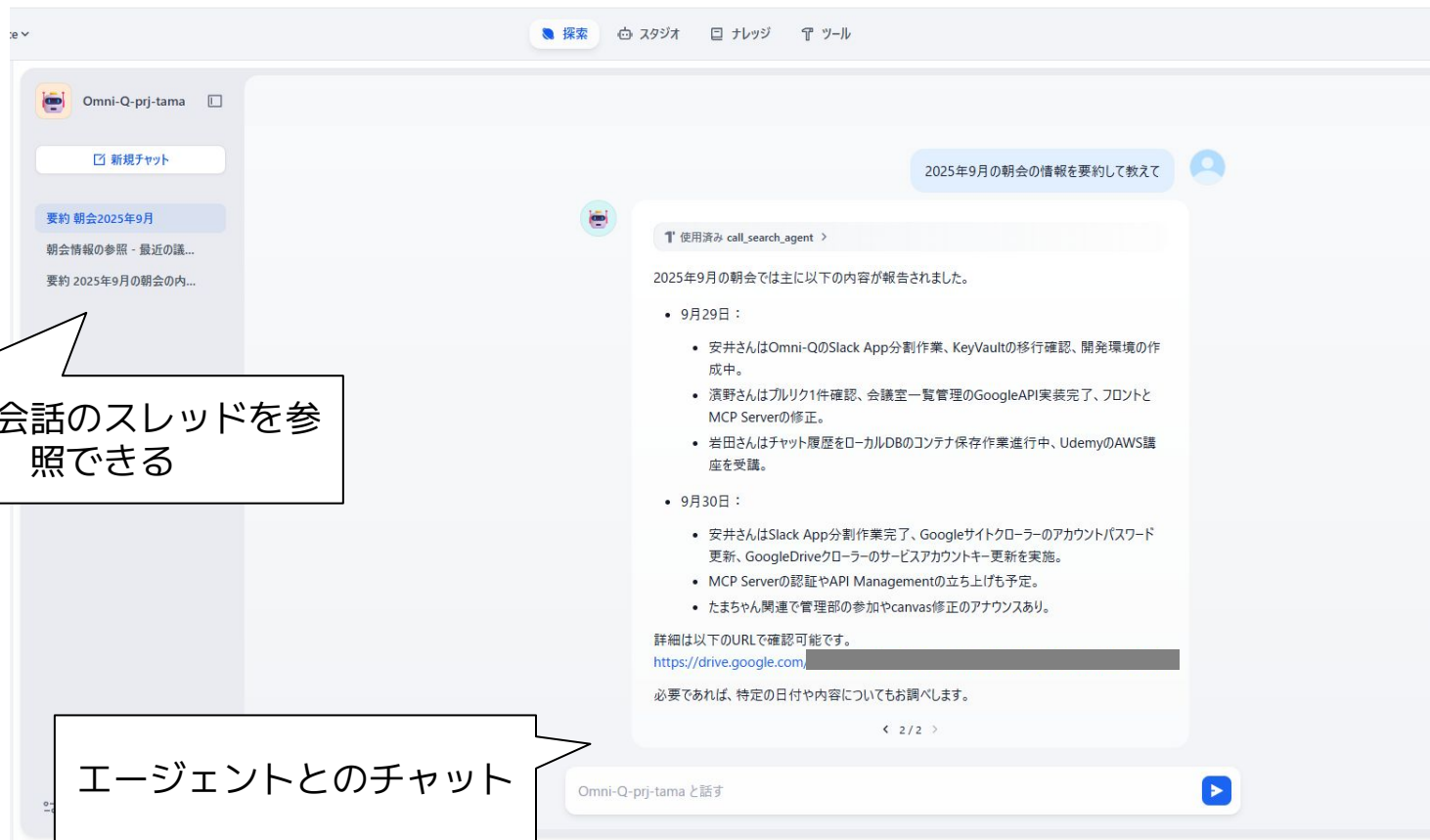
## エージェント作成画面

The screenshot displays the Dify AI agent configuration page. On the left, a sidebar lists navigation options: 'Omni-Q-prj-tama エージェント', 'ワークスペース', 'API アクセス', 'ログ & 注釈', and '監視'. The main area is titled 'ワークスペース' and contains several configuration sections: 'プロンプト' (Prompt), '変数' (Variables), 'コンテキスト' (Context), 'メタデータフィルタ' (Metadata Filter), and 'ツール' (Tools). The 'プロンプト' section is highlighted with a callout box stating 'エージェントに設定するプロンプトを書くエリア' (Area to write the prompt for the agent). The 'ツール' section shows a selected tool 'Prj-tama AI Search MCP call\_search\_agent' with a toggle switch, accompanied by a callout box stating 'Azure Functionsで作成したMCP Serverと連携' (Connect to the MCP Server created with Azure Functions). On the right, the 'デバッグとプレビュー' (Debug and Preview) section shows a chat interface with a callout box stating '作成したエージェントとチャットして動作確認' (Chat with the created agent to verify operation). The chat history shows a message about summarizing meeting minutes from October.

作成したエージェントとチャットして動作確認

エージェントに設定するプロンプトを書くエリア

Azure Functionsで作成したMCP Serverと連携

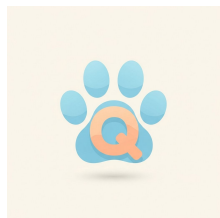


## エラーの解決方法を過去の情報から探す例

Azure上のリソースの設定変更時にSSL証明書検証エラー（self signed certificate in certificate chain）が発生します。WSLのAzure CLIにCatoの設定が必要かと考えていますが、過去に類似事例はありますか？



ユーザ



Omni-Q

Azure上のリソース設定変更時に発生するSSL証明書検証エラー（self signed certificate in certificate chain）について、過去にWSLのAzure CLIでCato証明書の設定が必要な事例がありました。具体的には、WSL上のAzure CLIが使っているPython環境のcertifiのcacert.pemファイル（例：/opt/az/lib/python3.11/site-packages/certifi/cacert.pem）の末尾にCatoの証明書を追加することでエラーが解消しています。こちらの設定はPythonでCato経由HTTPS接続をする場合にも必要で、Cato証明書が設定されていないとself signed certificate in certificate chainのエラーが発生します。設定参考資料や関連会議資料もあり、実際に設定してエラーが解消した報告があります。

- [具体的な手順例](#)
- [参考URL](#)：
- [Slack議論](#)：



## プロジェクトの週次報告資料作成を補助する例

2025年9月の朝会資料から、各人の作業内容を要約してください。



ユーザ

2025年9月の朝会情報から各人の作業内容を要約します。

【Aさん】

- Slack Appを分割し、開発環境・本番環境での移行確認と反映作業を実施。
- GoogleDriveクローラーをAzure Functions化し、サービスアカウントの管理。
- KeyVault接続設定の完了。

【Bさん】

- GoogleAPIを活用した会議室一覧管理をほぼ完了。
- AWS移行に向けてデプロイ手順検討やCloudFormationテンプレート作成中。
- プルリクエストの確認や資料作成（勉強会、MTG関連）

【Cさん】

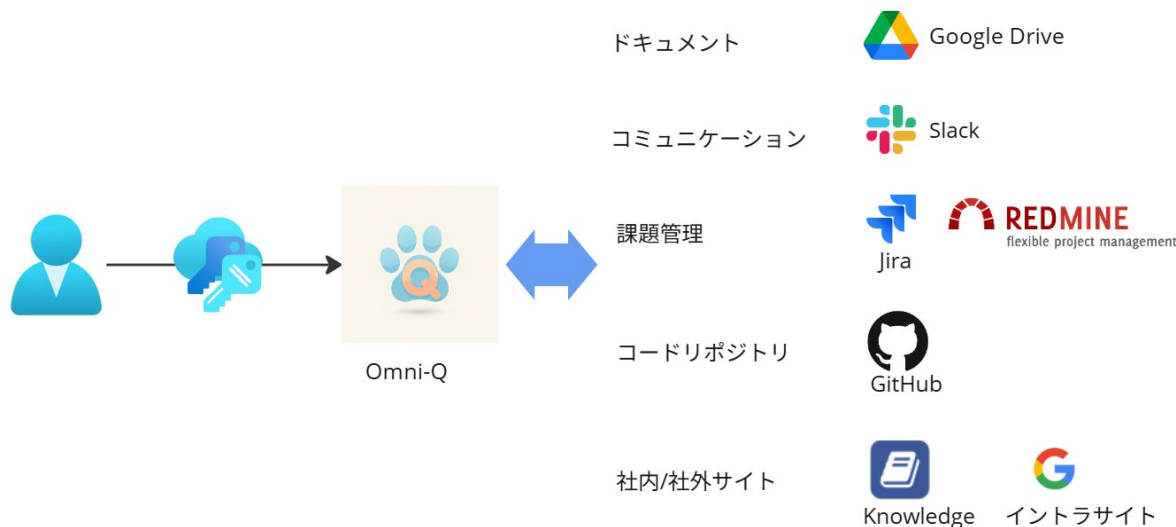
- チャットの履歴をローカルDBのコンテナに保存する開発を継続中。
- UdemyのAWS学習進行中

詳細は[こちら](#)の議事録ファイルにて確認できます



Omni-Q

- ◆ 現在6件のプロジェクト/コミュニティでベータテストを実施中
- ◆ フィードバックをいただきながら、より実践的なツールへと改善していく
  - Difyでの提供により自由にエージェントを作成可能にする
  - MCP Serverの認証の強化（Google認証など）
  - 情報源の追加（課題管理ツール、コード管理ツールなど）





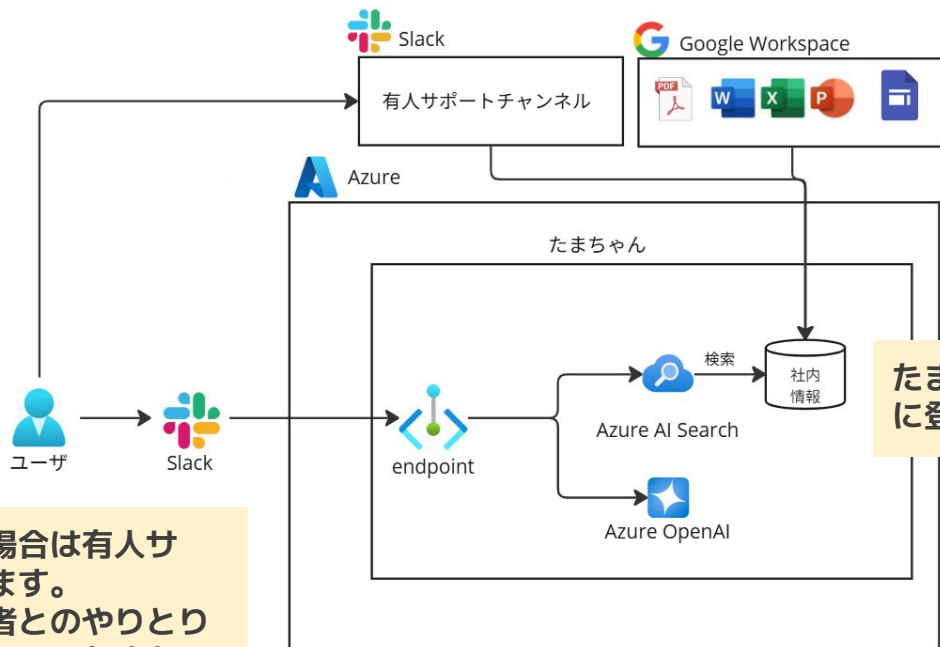
## ご清聴ありがとうございました

たまちゃんをはじめとするエクサ社内での生成AI活用の取り組みについては[社外向けHPのコラム](#)やDPSCイントラにて発信していますのでぜひご覧ください。



# Appendix

- ◆ たまちゃんは、Googleドライブ、Googleサイト、Slackの情報を事前に取り込み
- ◆ ユーザからの質問に対し、関連性の高い社内情報を検索し、その内容に基づいて回答を生成します (RAG: Retrieval Augmented Generation)



たまちゃんが参照する社内文書は事前に登録しています

たまちゃんが回答できない場合は有人サポートチャンネルを案内します。案内先チャンネルでの担当者とのやりとりをたまちゃんに取り込むことで、たまちゃんの精度向上を図ります。

- ◆ たまちゃんが参照する情報（コンテンツ）を管理・更新するための機能を提供しています
- ◆ 各コンテンツの登録、更新、削除といった運用は、それぞれの情報を管轄する部署の情報オーナーが担当します

クローラー管理画面

bot識別  
たまちゃん

組直で絞り込み  
DPSC

### GoogleSiteクローラー設定

クローラーに登録

URL  タグ名  個人情報 (PII) マスク

クローラーに登録されているサイト一覧

ボットID	URL	サイト名 (クローリング後に取得)	タグ名	個人情報 (PII) マスク	ステータス	実行時間	
tama-chan	https://sites.google.com/exa-corp.co.jp/exa-intratop/	exa-portal - たまちゃん 社内サービス	DPSC	オフ	成功	2024-07-31 23:00:02 JST	<input type="button" value="削除"/>
chan	https://sites.google.com/exa-corp.co.jp/slap4	slap4サービス	DPSC	オフ	成功	2024-07-31 23:00:02 JST	<input type="button" value="削除"/>

### GoogleDriveクローラー設定

クローラーに登録

# 管理者向け機能②：利用状況の分析

- ◆ たまちゃんの月ごとの利用状況を可視化し、分析するための機能を提供しています
- ◆ 月別アクティブユーザ数、総利用回数などを確認することで、利用実態を把握し、サービス品質の向上や利用促進に繋がります

bot選択  
たまちゃん

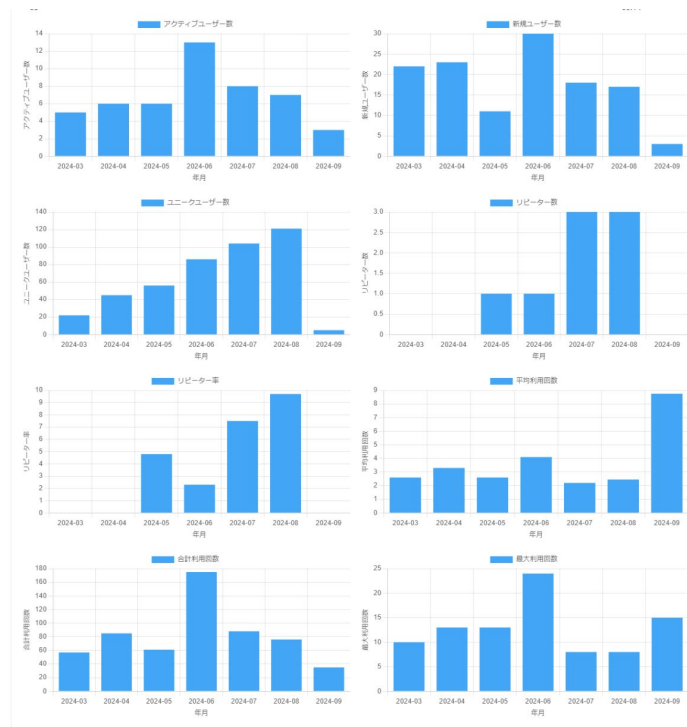
利用状況 答えられなかった質問

From 年  
2024

From 月  
3

To 年  
2024

年月	アクティブユーザ数	新規ユーザー数	ユニークユーザー数	リピーター数	リピーター率	平均利用回数
2024-09	10	16	137	5	12.8	3.6
2024-08	7	17	121	3	9.7	2.45
2024-07	8	18	104	3	7.5	2.2





## 1. 評価データセットの準備

客観的な精度評価のため、多様な質問とそれに対応する模範解答のペアを複数用意します



## 2. 評価指標の定義・実装

検索精度、回答精度、全体としての正解率を計測します



## 3. 評価

準備した評価データセットと定義済み指標に基づき、たまちゃんの精度を評価します



## 4. 改善施策の検討

評価結果から改善点を検討します

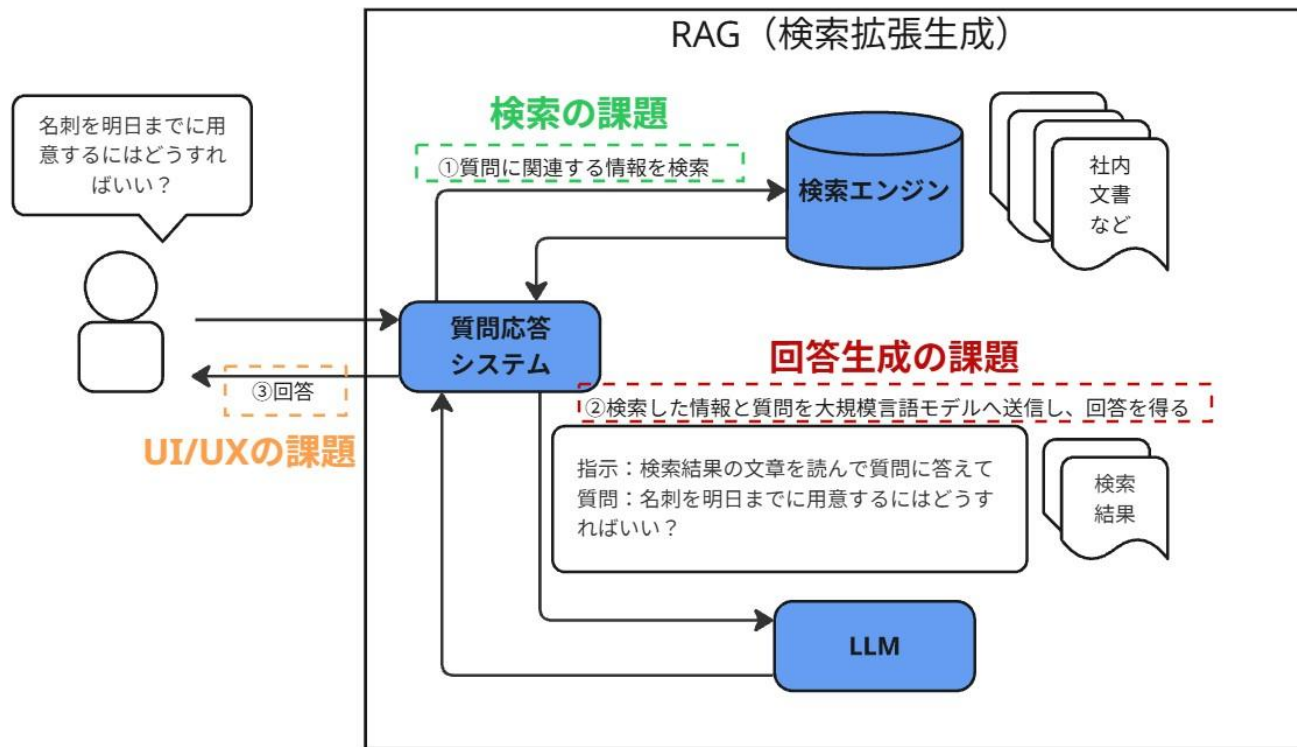


## 5. 改善施策の実施

精度向上のための具体的な施策を実行します

※3～5の改善サイクルを継続的に回すことで、たまちゃんの回答精度を向上させることを目指しています

検索・回答生成・UI/UXの3つの課題それぞれについて改善に取り組みました



施策	課題の種類	回答精度の向上度合	導入コスト	補足
同義語辞書の登録	検索の課題	低	中	<ul style="list-style-type: none"><li>検索精度の向上には貢献したが、検索できても同義語をLLMが知らないため回答できないケースが残る</li><li>辞書を準備する手間がかかる</li></ul>
上位3件から5件へ拡張		低	低	
セマンティックランカー		中	低	<ul style="list-style-type: none"><li>Azure AI Search限定機能</li><li>並び替え機能のみ使用 &amp; インデックス設計の見直しが不要なケースでは、容易に導入可能</li></ul>
想定質問の追加		中（※）	大	<ul style="list-style-type: none"><li>（※）セマンティックランカーと併用する場合、回答精度向上には寄与しなかったので注意</li><li>使用する全文検索エンジンにおいて、セマンティックランカーが利用できないケースに有効</li></ul>
プロンプト構造見直し	回答生成の課題	低	低	<ul style="list-style-type: none"><li>検索に成功していても、プロンプトの品質が低いと回答生成に失敗しやすい</li></ul>



# 施策ごとの回答精度と導入コストのサマリー

施策	課題の種類	回答精度の向上度合	導入コスト	補足
chunkサイズの調整	検索の課題	中	中	<ul style="list-style-type: none"><li>適切なチャンクサイズに設定することで正解率の向上と回答内容の改善が見られた</li><li>チャンクサイズを大きくすると料金が上がる点に注意が必要</li></ul>
ベクトルモデル最新化		低	中	<ul style="list-style-type: none"><li>旧モデルと新モデルで、ベクトルに互換性がないため、インデックスの再作成が必要</li></ul>
表のMarkdown化	回答生成の課題	低	中	<ul style="list-style-type: none"><li>表をMarkdown形式にすることで回答できるようになった質問がある一方で、検索で上手くドキュメントがヒットしなくなり、回答できなくなった質問も存在</li></ul>
回答生成時に同義語参照		低	中	<ul style="list-style-type: none"><li>全体としての精度は変わらなかったが、同義語を参照することで答えられるようになったものが一部あった</li><li>辞書を準備する手間がかかる</li></ul>

# 施策ごとの回答精度と導入コストのサマリー

施策	課題の種類	回答精度の向上度合	導入コスト	補足
回答生成時に o3-mini	回答生成の課題	低	低	<ul style="list-style-type: none"><li>回答速度は低下するが、回答精度が向上し、利用料金も削減</li></ul>
回答生成時に GPT-4.1		中	低	<ul style="list-style-type: none"><li>利用料金はそのままに、回答速度と回答精度が向上</li><li>「検索クエリ/タグ生成」と「想定質問生成」において、GPT-4.1 miniを検証したが、現行のGPT-4o miniと比べて回答精度が向上せずに利用料金だけが増加してしまうことが分かった</li></ul>

—EOF—